ELSEVIER

Image and Vision Computing 20 (2002) 307–318

image and vision COMPUTING

www.elsevier.com/locate/imavis

Corresponding dynamic appearances

Shaogang Gong[a,*], Alexandra Psarrou[b], Sami Romdhani[b]

[a]*Department of Computer Science, Queen Mary, University of London, London E1 4NS, UK*
[b]*Harrow School of Computer Science, University of Westminster, Harrow HA1 3TP, UK*

Received 16 October 2000; received in revised form 15 October 2001; accepted 2 January 2002

Abstract

Modelling the appearance of 3D objects undergoing large pose variation relies on recovering correspondence of both shape and texture across views. The problem is hard because changes in pose not only introduce self-occlusions hence inconsistent 2D features between views, but also cause non-linear variations in both the shape and texture of object appearance. In this paper, we present an approach for establishing structured sparse correspondence between face images across views using non-linear shape models. We extend the non-linear shape models to dynamic appearance models of both shape and texture across views. For non-linear model transformation, we adopt Kernel PCA. For bootstrapping appearance alignment at different views, we introduce a generic-view shape template. We show that Kernel PCA constrained the dynamic appearance model and eases model fitting to novel images. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: View-based representation; Appearance models; The correspondence problem; Active shape models; Support vector machines; Kernel principal components analysis

1. Introduction

It has been shown in recent years that it is possible to faithfully represent the 3D structure of an object with only 2D views of the object. This is not only psychophysically more plausible but also computationally more advantageous over conventional 3D models [30,33,34]. Such a multi-view based 2D representation, however, assumes that dense correspondence between views can be established. This can be non-trivial especially when an object such as a human face appears in different views or when faces appear differently due to the change of identity. Ambiguities often exist in defining correspondence for all points in the images of different appearances of an object, even at a single view. For objects such as human faces, although the overall structure of two faces is likely to be the same (i.e. two eyes and one nose above a mouth), the fine scale structure can differ significantly. Even though dense correspondence cannot always be defined, attempts have been made and the methods tend to rely on the computation of optical flow. They are inevitably computationally expensive and usually require human intervention in order to ensure acceptable performance [2,36]. Due to the difficulty and the likely prohibitive cost from computing dense correspondence, these models are mainly designed for face synthesis and applications in computer graphics.

Correspondence can be made more consistent and robust at a sparse level where there is common structure [17]. In particular, sparse correspondence between a carefully chosen set of 2D feature points of an object can be more reliable if the feature points are selected to form the salient structure, i.e. *shape* of the object. For example, at a fixed view such as frontal view, the non-rigid 2D shape of a face can be modelled using a *linear active shape model* based on a set of facial salient feature points and their local grey-levels [7,9,23]. Fig. 1 shows different selections of facial feature points adopted for correspondence by various face models proposed in recent years.

In the presence of large view variation, self-occlusion unfortunately restricts such linear shape models to a single or narrow view. This is because a face undergoing rotation in depth (change in view or pose)[1] results in a significantly non-linear transformation of its shape in the image space. To overcome this problem, there have been a number of solutions proposed largely based on explicit 3D pose estimation and piecewise view-based models. For example, geometric

[*] Corresponding author. Tel.: +44-207-882-5249; fax: +44-208-980-6533.

E-mail address: sgg@dcs.qmul.ac.uk (S. Gong).

[1] In this article, we refer *view-sphere* as a collage of views spanned by continuous variations in both yaw and tilt of a human face. In this context, *view* is also referred to as *pose*. In the rest of this article, we are more concerned with pose or view change caused by yaw variation which is far more significant than that of tilt.

PII: S0262-8856(02)00025-2

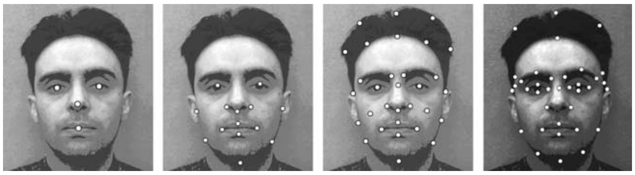

Fig. 1. Different salient facial feature points selected for establishing correspondence between face images. From left to right, correspondence is established between four [29], 14 [22], 31 [38] and 36 [11] points, respectively.

models were exploited for pose estimation through utilising a 2D affine-model of image positions of the mouth and eyes and by adding the nose-tip position across views [15,16]. Alternative models for pose estimation have also been introduced based on affine transformation of holistic templates [24], and by measuring similarity to prototype views [19,32]. Examples of piecewise view-based models include Kruger et al. [21] who used elastic graph matching of connected nodes of Gabor filter 'jets' to locate and estimate face pose. Different graph models were used for different poses, leading to a computationally expensive approach. Other piecewise models of multiple views have also been introduced using view-based eigenspaces [13,29], mixture models [8] or support vector machines (SVM) [27].

In this work, we describe a method for effectively corresponding a single non-linear dynamic face appearance model across views. In Section 2, we outline the need for corresponding active shape models (ASMs) across views before we introduce Kernel PCA as a technique for learning non-linear model transformations in Section 3. This is to extend the linear ASMs to non-linear dynamic shape models for establishing structured sparse correspondence across the view-sphere. In Section 4, we describe an effective algorithm for fitting a non-linear face shape model to novel images and simultaneously recover their poses. Such a shape model captures all possible 2D shape variations in a training set and performs a non-linear model transformation during fitting. In Section 5, we further extend the non-linear shape model to a dynamic appearance model of both shape and texture. We introduce a generic-view shape template for bootstrapping the alignment of dynamic appearances at different views. Kernel PCA is extended for non-linear model transformation of both shape and texture arising from large pose variations and the regression required for model fitting is eased as a result. Experiments are shown in Section 6 before we conclude with possible future work in Section 7.

2. Shape correspondence at a narrow view: Active shape models

The eventual role of correspondence in a view-based representation is to bring not only the shape but also the appearance of an object into *alignment*[2] [3,13,37]. To this end, one needs to model both structural (shape) and statistical (texture) knowledge about the object. Statistical knowledge can only be effectively exploited if structural constraints are sufficiently satisfied through establishing correspondence. Beymer [1] used a shape representation in which dense correspondence is required for all the pixels of a face image before texture warping is performed. This is achieved by computing optical flow which registers *all* the pixels of a face with those of a mean face. The shape of the new face is modelled by displacement vectors from the shape of the mean face. This shape information is then used for performing a simple 2D warping in order to generate its *shape-free texture* [12]. When a novel face image is presented to the model, the shape and the texture of the face are recovered based on an iterative optimisation fitting algorithm. The accuracy of this feature alignment obviously depends on the optical flow estimation and it is necessarily expensive. To overcome the problem, sparse correspondence can be adopted. However, in order to constrain the degrees of freedom in corresponding a set of sparse salient feature points extracted from different images, one should avoid corresponding every pair of individual feature points independently [13,28]. To this end, prior knowledge on any plausible shapes that can be formed by the feature set is needed in order to establish holistic shape based correspondence. It is also desirable if such prior knowledge can be learned from examples. This is the essence of linear ASMs.

Cootes et al. [7,9,23] have shown extensively that the 2D shape of objects can be effectively modelled using linear ASMs. Such linear shape models have also been extended to facilitate representation of appearance (both shape and texture) change using *active appearance models* (AAM) [6]. An ASM consists of a point distribution model (PDM) aiming to learn the variations of valid shapes based on a set of salient feature points (landmarks) that best represent the shape of an object, and a set of local models of grey-levels around these landmarks. For model building, a set of training images are warped into a mean shape which brings the landmarks into alignment. A shape-free texture vector can then be obtained by 2D warping using methods such as the Bookstein's algorithm based on thin plate splines [22] or linear interpolation [10]. While the accuracy of this warping is precise for the landmarks, it is only approximate for all other pixels in between and their accuracy depends on the number of landmarks used and their relative positions. After warping to the mean shape, each training example image can be represented as a shape vector and a shape-free texture vector [6,12]. The elements of the shape vector are the 2D image co-ordinates of the landmarks. The elements of the texture vector are the intensity values of the warped, shape-free image pixels. The PDM and local grey-level models are then learned using these shape and texture vectors as examples. The computational difficulty now is to perform model fitting which requires on-line correspondence to be established between a model and a novel image. The use of a mean shape avoids the need to compute many

[2] Here alignment does not only limit to linear cases.

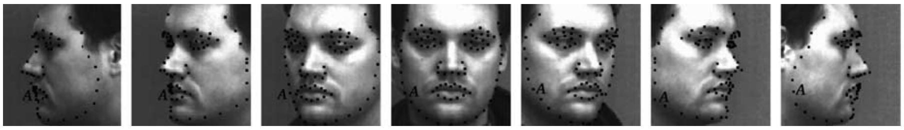

Fig. 2. The 2D shape of a face across views from −90° to +90° is given by a set of 80 facial landmarks and their corresponding local grey-levels. However, the local grey-levels around the landmarks vary widely in this case. This is highlighted for one of the landmarks $\mathscr{A}$, which clearly cannot be established across views solely based on local grey-levels.

correspondence maps. However, matching still entails a relatively expensive search over parameters of variations in shape and texture. The match can be based on models of the local appearance of the landmarks [38] or along curves joining the landmarks [14,22,25].

Crucially, linear ASMs are based on a number of implicit assumptions: (a) the shape of the object of interest can be defined by a relatively small set of explicit view models, (b) the grey levels around a particular landmark are consistent for all the views of the object and can be used to find correspondences between these views and, (c) the shapes at different views vary linearly. However, assumptions (b) and (c) can be easily violated when shape variations are caused by significant changes in pose, as illustrated by the example in Fig. 2.

Hence, whilst ASM can be used to model and recover some changes in the shape of an object, it can only cope with largely linear variations. When the valid shape region (VSR) in the shape space is non-linear, as in the case when large pose variations are allowed, the PDM of an ASM requires non-linear transformations. The problem can be partially addressed using combination of linear components [8,20]. For instance, a single ASM was shown to cope with shape variations from a narrow range of face poses (turning and nodding of ±20°). Non-linear variations caused by changes in pose and self-occlusions can be captured using the combination of five different linear models [23]. However, the use of linear components not only increases the dimensionality of the model but also can potentially introduce invalid shape variations [5,28]. Although the dynamics of valid non-linear model transformation can be captured by a set of structured linear models using non-linear principal components analysis (PCA) such as cluster based on-linear principal component analysis [4] and hierarchical PCA [20,28]. This approach, however, does require a rather large database for learning the distribution of the linear subspaces. An alternative approach is Kernel PCA.

3. Learning non-linear transformation in model space: Kernel PCA

Kernel principal components analysis (KPCA) is a non-linear PCA method recently introduced by Schölkopf et al. [31] based on SVM [35]. The essential idea of KPCA is both intuitive and generic. In general, PCA can only be effectively performed on a set of observations that vary linearly. When the variations are non-linear, they can always be mapped into a higher dimensional space, which is again linear. If this higher dimensional linear space is referred to as the *feature space* ($\mathscr{F}$), KPCA utilises SVM to find a computationally tractable solution through a simple kernel function which intrinsically constructs a non-linear mapping from the input space to $\mathscr{F}$. As a result, KPCA performs a non-linear PCA in the input space.

More precisely, if a PCA is aimed at decoupling non-linear correlations among a given set of shape vectors $\mathbf{x}_j$ through diagonalising their covariance matrix, the covariance can be expressed in a linear feature space $\mathscr{F}$ instead of the non-linear input space, i.e.

$$\boldsymbol{C} = \frac{1}{M}\sum_{j=1}^{M} \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^{\mathrm{T}} \tag{1}$$

where $\Phi(\cdot)$ is a non-linear mapping function which projects the input vectors from the input space to the $\mathscr{F}$ space. To diagonalise the covariance matrix, the eigen-problem $\lambda\mathbf{p} = \mathbf{Cp}$ must be solved in the $\mathscr{F}$ space. As $\mathbf{Cp} = (1/M)\sum_{j=1}^{M}(\Phi(\mathbf{x}_j)\cdot\mathbf{p})\Phi(\mathbf{x}_j)^{\mathrm{T}}$, all non-singular solutions $\mathbf{p}$ with $\lambda \neq 0$ must lie in the span of $\Phi(\mathbf{x}_1),\ldots,\Phi(\mathbf{x}_M)$. This eigen-problem is equivalent to

$$\lambda(\Phi(\mathbf{x}_k)\cdot\mathbf{p}) = (\Phi(\mathbf{x}_k)\cdot\mathbf{Cp}) \tag{2}$$

for all $k = 1,\ldots,M$ and there exists coefficients α_i such that

$$\mathbf{p} = \sum_{i=1}^{M} \alpha_i \Phi(\mathbf{x}_i). \tag{3}$$

Substituting Eq. (2) with Eqs. (1) and (3) gives

$$\lambda\sum_{i=1}^{M}\alpha_i(\Phi(\mathbf{x}_k)\cdot\Phi(\mathbf{x}_i))$$

$$= \frac{1}{M}\sum_{i=1}^{M}\alpha_i\left(\sum_{j=1}^{M}(\Phi(\mathbf{x}_k)\cdot\Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_j)\cdot\Phi(\mathbf{x}_i))\right) \tag{4}$$

It is important to note that this eigen-problem only involves dot products of mapped shape vectors in the feature space $\mathscr{F}$. This is the raison d'être of this method. Indeed, the nature of structural risk minimisation (SRM) suggests that mapping $\Phi(\cdot)$ may not always be computationally tractable even if it exists [35]. However, it needs not be explicitly computed if SRM is implemented using SVMs. Only dot products of two vectors in the feature space are needed. Even so, since the feature space has high dimensionality,

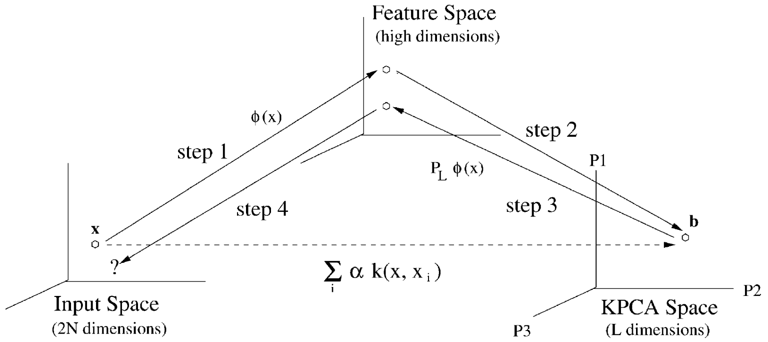

Fig. 3. *Conceptually*, KPCA performs a non-linear mapping $\Phi(\mathbf{x})$ to project an input vector to a higher dimensional feature space $\mathcal{F}$ (step 1). A linear PCA is then performed in this feature space giving a lower dimensional KPCA space based representation (step 2). To reconstruct an input vector from the KPCA space, its KPCA representation is projected into the feature space (step 3) before an inverse $\Phi(\mathbf{x})$ mapping is performed (step 4). *Computationally*, however, none of the four steps is performed. The mapping is in fact carried out directly by kernel functions $\sum_i \alpha k(\mathbf{x}, \mathbf{x}_i)$ between the input space and its KPCA space, shown as the dashed line in the diagram. For reconstruction, this kernel-based mapping is only approximated. Optimisation is required in the KPCA space in order to find a best match between the model and the KPCA representation of the input vector.

computing such dot products could still be computationally expensive if at all possible. An SVM can be used to avoid explicitly performing either mappings $\Phi(\cdot)$ or the dot products in the high dimensional feature space $\mathcal{F}$.

Let us define an $M \times M$ matrix $\mathbf{K}$ where $k_{ij} = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$, Eq. (4) can then be rewritten as

$$M\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha} \tag{5}$$

where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_M]^{\mathrm{T}}$. Now, performing PCA in the feature space $\mathcal{F}$ amounts to resolving the eigen-problem of Eq. (5). This yields eigenvectors $\boldsymbol{\alpha}^1, ..., \boldsymbol{\alpha}^M$ with eigenvalues $\lambda^1 \geq \lambda^2 \geq \cdots \geq \lambda^M$. Dimensionality can be reduced by retaining only the first L eigenvectors. The principal components $\mathbf{b}$ of a shape vector $\mathbf{x}$ are then extracted by projecting $\Phi(\mathbf{x})$ onto eigenvectors $\mathbf{p}^k$ where $k = 1, ..., L$

$$b_k \equiv \mathbf{p}^k\cdot\Phi(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i^k(\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x})) \tag{6}$$

To solve the eigen-problem of Eq. (5) and to project from the input space to the KPCA space using Eq. (6), one can avoid computing either the dot products in the feature space or the mappings through constructing a SVM (Fig. 3). This is achieved by finding a *kernel function* when applied to a pair of shape vectors in the input space, it yields the dot product of their mapping in the feature space

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x})\cdot\Phi(\mathbf{y}) \tag{7}$$

There exists a few kernel functions which satisfy the above criterion [35]. This includes the Gaussian kernel where $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/2\sigma^2)$. We adopted the Gaussian kernel function after experimental evaluation which concluded that it performed better than the step function. This SVM kernel function effectively provides a low dimensional Kernel-PCA subspace which represents the distribution of the mapping of the training vectors in the high dimensional feature space $\mathcal{F}$. As a result, non-linear transformation in the input space can be performed by reconstructions from the KPCA subspace. However, this process can be problematic [26]. The vectors in the feature space $\mathcal{F}$ which have a pre-image in the input space are those that can be expressed as a linear combination of $\Phi(\mathbf{x}_1), ..., \Phi(\mathbf{x}_M)$. However, if the reconstruction in $\mathcal{F}$ is not perfect, there is no guarantee to find a pre-image of the reconstruction in the input space (Fig. 3). Especially if dimensionality reduction is applied, the reconstruction from the KPCA space to $\mathcal{F}$ is only an approximation. Therefore the reconstruction ($\hat{\mathbf{x}}$) of an input vector ($\mathbf{x}$), whose principal components are truncated to the first L components, is approximated by minimizing

$$\|\Phi(\hat{\mathbf{x}}) - P_L\Phi(\mathbf{x})\|^2 \tag{8}$$

where P_L is a truncating operator. To solve this optimisation problem, there exists techniques tailored to particular kernels [26].

4. Shape correspondence across views: Dynamic shape models

Existing ASMs of faces exhibit only limited pose variations due to its linearity. One implicit but crucial assumption of the existing method is that correspondences between landmarks of different views can be established solely based on the grey-level information. However, when large non-linear shape variations are introduced due to changes in object pose, local grey-level values around landmarks are also view-dependent.

In the case of face varying from profile to profile, the strongest contextual information is given by pose. Hence, we augment the shape vector in PDM using pose θ, i.e. $(x_1, y_1, ..., x_N, y_N, \theta)$ where (x_i, y_i) are the coordinates of the ith landmark. Similarly, the model for the local grey-levels (LGLs) around each landmark is a concatenation of the grey-levels along the normal to the shape contour and

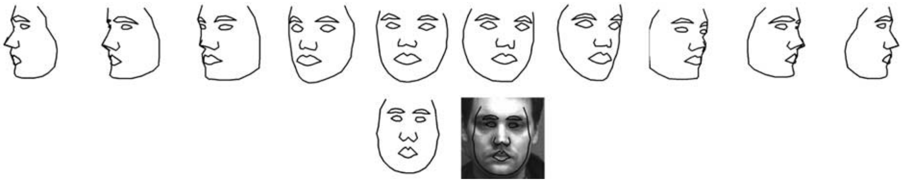

Fig. 4. Top: example of training shapes. Bottom: the mean shape at frontal view.

the pose of the face. It is worth pointing out though that in general, care needs to be taken, through appropriate weighting, in the construction of a hybrid representation using different sub-components. This is especially true if the distribution variance among different sub-spaces is very large [5,28]. Both the PDM and the LGLs are built using KPCA. Model fitting is an iterative process starts from the frontal view of the shape located near the object in the image. Notice that it is better to start from a specific view rather than the mean shape, as was adopted by Cootes et al. [9]. This is because due to large shape variations, the mean shape across views is no longer a valid shape, as shown in Fig. 4.

To start model fitting, it is assumed that a rough position of a face in the image is known. However the pose is unknown and the fitting process recovers both the shape of the face and its pose. The computation is performed as follows:

1. To find plausible correspondences of landmarks between views, augmented local grey-level models are used. To this end, the KPCA reconstruction of the grey-level vector is minimised along the normal to the shape. To compute the KPCA reconstruction of a vector, one first projects this vector to the KPCA space using Eq. (6), obtaining the kernel principal components (**b**). The reconstruction is then performed by minimising the norm given in Eq. (8). During the first iteration the pose of the object is unknown therefore the reconstruction error must also be minimised with respect to poses. This process yields an estimation of both the landmark loci and the pose for each landmark. The newly estimated pose is then the average pose of all the landmarks. This pose is to be used to constrain the shape within the VSR in step 3.
2. The estimated shape is aligned following Cootes et al. [9].
3. To constrain the estimated shape within the VSR, it is projected to the shape space using a pose augmented non-linear PDM given by Eq. (6), constrained to lie within the VSR by limiting the values of **b** [9] and projected back to the input space using Eq. (8). This yields a new estimated shape. Its pose will be used to locate the correspondence of the landmarks in the next iteration. Repeat step 1 until convergence.

5. Learning to transform dynamic appearances across views

We now extend the non-linear shape models to appearance models across views with which both non-linear shape and texture deformations are modelled using KPCA. First, let us introduce the notion of a *generic-view 2D shape template* for bootstrapping appearance alignment at different views for training. In principle, the problem of missing (hidden) features between different views due to self-occlusion can be addressed in two ways. The hidden features can be reconstructed using the information of the visible features. This method may not be feasible without resorting explicitly to 3D information. Alternatively, the hidden features can be explicitly represented by a generic-view 2D shape template without regenerating their texture.

Suppose a shape $\mathbf{X}$ is composed of a set of N_s landmarks $\mathbf{x}_i$ and the texture $\mathbf{v}$ is composed of N_t grey-level values v_i

$$\mathbf{x}_i = [x_i\ y_i]^{\mathrm{T}}, \qquad \mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_{N_s}]^{\mathrm{T}}, \qquad v = [v_1 \cdots v_{N_t}]^{\mathrm{T}} \tag{9}$$

To bootstrap the alignment process at different views, a generic-view 2D shape template, denoted by $\mathbf{Z}$, is introduced with which the landmarks are to be aligned view-dependently. The shape of any single view $\mathbf{X}$ is composed of two types of landmarks: (a) $\mathbf{X}_{\mathrm{out}}$, the outer landmarks which define the contour of the face and (b) $\mathbf{X}_{\mathrm{in}}$, the inner landmarks which define the position of facial features such as mouth, nose, eyes and eyebrows. The generic-view shape template $\mathbf{Z}$ is computed based on M training shapes

$$\begin{aligned} &\mathbf{Z}_{\mathrm{in}} = G_{\mathrm{in}}(\mathbf{X}_1, \ldots, \mathbf{X}_M), \qquad \mathbf{Z}_{\mathrm{out}} = G_{\mathrm{out}}(\mathbf{X}_1, \ldots, \mathbf{X}_M), \\ &\mathbf{Z} = \mathscr{H}(\mathbf{Z}_{\mathrm{out}}, \mathbf{Z}_{\mathrm{in}}) \end{aligned} \tag{10}$$

where $\mathscr{H}(\cdot)$ is an operator which concatenates an outer shape and an inner shape, yielding a complete shape. Once $\mathbf{Z}$ is obtained, the shape $\mathbf{X}$ and its associated texture $\mathbf{v}$ can be aligned, giving an aligned shape ($\hat{\mathbf{X}}$) and its corresponding aligned texture ($\hat{\mathbf{v}}$)

$$\hat{\mathbf{X}} = F(\mathbf{X}, \mathbf{Z}), \qquad \hat{\mathbf{v}} = W(\mathbf{v}, \hat{\mathbf{X}}, \mathbf{X}) \tag{11}$$

After this initial alignment, shape transformation is required in order to align all inner landmarks at different views. This is because as pose changes, the outer landmarks, which define the contour of the shape, would 'travel' on the face. Therefore, the alignment error e_i of face i only takes into account the inner landmarks

$$e_i = \|\mathbf{Z}_{\mathrm{in}} - \hat{\mathbf{X}}_{\mathrm{in}}\|^2 \tag{12}$$

In general, the minimisation of this generic error function

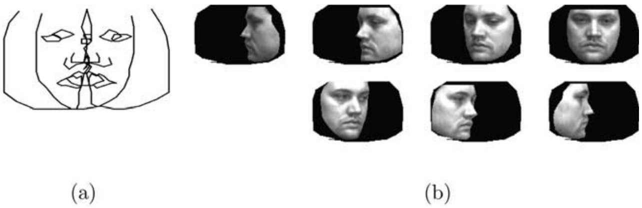

Fig. 5. (a) A generic-view shape template is based on aligning shapes from two profiles and the frontal view with respect to a single or a set of inner landmarks. (b) Examples of aligned appearances of both shape and texture at different poses using just one inner landmark assigned to the nose-tip.

can be computationally expensive with no guarantee of convergence. To avoid this problem, we model this non-linear transformation indirectly through learning using KPCA.

Let us first compute the generic-view shape template for the initial alignment of the training data using the following $F(\cdot)$, $W(\cdot)$, $G_{\text{in}}(\cdot)$ and $G_{\text{out}}(\cdot)$ functions:

(1) The training shapes at each view are scaled according to the distance between the chin-tip and the nose-tip in the generic-view shape template (Fig. 5). The shapes at each view are aligned with respect to the nose-tip by translating and scaling individual landmarks of a shape $\mathbf{X}$ to $\tilde{\mathbf{X}}$

$$\tilde{\mathbf{x}}_i = \frac{(\mathbf{x}_i - \mathbf{x}_k)}{\|\mathbf{x}_k - \mathbf{x}_l\|}, \qquad i = 1, \dots, N_s \tag{13}$$

where k refers to the landmark at the nose-tip and l to the chin-tip.

(2) These aligned shapes at different views are superimposed to form the generic-view 2D shape template using, in a simple case, the mean of the inner landmarks and the extreme outer landmarks of the M training shapes

$$\mathbf{Z}_{\text{in}} = G_{\text{in}}(\cdot) = \frac{1}{M}\sum_{i=1}^{M}(\tilde{\mathbf{X}}_{\text{in},i}), \;\; \mathbf{z}_{\text{out},j} = G_{\text{out}}(\cdot) = \tilde{\mathbf{x}}_{i,j},$$
$$\text{if } \tilde{\mathbf{x}}_{i,j} \not\subset \{\tilde{\mathbf{X}}_{\text{out},1} \cap \cdots \cap \tilde{\mathbf{X}}_{\text{out},M}\} \;\; \forall i = 1, \dots, M, \; j = 1, \dots, N_s \tag{14}$$

where $\tilde{\mathbf{x}}_{i,j} \not\subset \{\tilde{\mathbf{X}}_{\text{out},1} \cap \cdots \cap \tilde{\mathbf{X}}_{\text{out},M}\}$ if a point $\tilde{\mathbf{x}}_{i,j}$ on the face is not in the overlapped area of all the training shapes. This generic-view shape template can be formed by the full set of outer landmarks but only one inner landmark. The outer landmarks are the extreme landmarks of the superimposed shapes. In this simple case, the mean of the inner landmarks can also be simply replaced by the tip of the nose. In general however, instead of the mean, more inner landmarks can be used for increased robustness with added computational cost [13].

This initial alignment process is shown in Fig. 5. To align both the shape and texture to the generic-view shape template, a fast affine transformation is applied. For the simple case of one inner landmark assigned to the nose-tip, the aligned shape and texture are computed by simply performing scaling and translating according to the generic-view template following Eqs. (13) and (16). For a more general case when there are more than one inner landmark, we define the scaling and translation as

$$\hat{\mathbf{x}}_i = F(\cdot) = \frac{(\mathbf{x}_i - \mathbf{z}_i)}{\|\mathbf{z}_k - \mathbf{z}_l\|} \qquad \forall i = 1, \dots, N_s \tag{15}$$

$$W(\cdot) = \begin{cases} \hat{\mathbf{v}}_{\mathbf{p}_i} = \mathbf{v}_{\mathbf{p}_i} & \text{if } \mathbf{p}_i \subset \hat{\mathbf{X}} \\ \hat{\mathbf{v}}_{\mathbf{p}_i} = 0 & \text{if } \mathbf{p}_i \not\subset \hat{\mathbf{X}} \end{cases} \qquad \forall \mathbf{p}_i \in \mathbf{Z} \tag{16}$$

where $\mathbf{v}_{\mathbf{p}_i}$ denotes the grey-level value at pixel $\mathbf{p}_i$. Examples of alignment using just one inner landmark are shown on the right in Fig. 5.

To utilise the generic-view shape template, all feature points including the hidden features are made explicit to the model all the time: a special value of grey-level is used to denote hidden points (0 or black). However, the hidden points are only approximate. For instance at 50° view, the hidden points behind the bridge of the nose are not treated. In addition, the alignment performed is coarse: $e_i = 0$ is only true for the nose-tip. The other features are only approximately aligned as illustrated in (a) Fig. 6. Once the initial alignment is performed, non-linear model transformations defined as the minimisation of the error function in Eq. (12) are entailed through learning using KPCA.

To summarise, our process for learning a *dynamic appearance model* of both shape and texture across views is illustrated in (b) of Fig. 6. This is achieved by (a) a generic-view shape template for bootstrapping alignment at different views, (b) KPCA based learning of non-linear model transformation across views and (c) KPCA constrained model fitting using simple linear regression. This dynamic appearance model extends the active appearance model introduced by Cootes et al. [6] to non-linear variations across views.

It is worth pointing out, however, that the reconstruction of a vector from the KPCA space to the original space requires to solve an optimisation problem that is computationally expensive [26]. This problem can be solved by an iterative algorithm whose solution is heavily dependent on the vector used to start the computation. In other words, the reconstruction of KPCA in the input space can only be obtained if a good approximation of that reconstruction is given. This assumption is appropriate when KPCA is applied to pose augmented non-linear transformation of shape models as described in Section 4. However, for fitting a dynamic appearance model of both shape and texture across views, an approximation of reconstruction in the image space is not available, except for the first iteration. Such an approximation is necessary if the KPCA based reconstruction is to converge. To overcome this problem, we adopt a simple scheme by which the reconstruction process iterates based on both the current parameters and the reconstruction of the previous iteration.

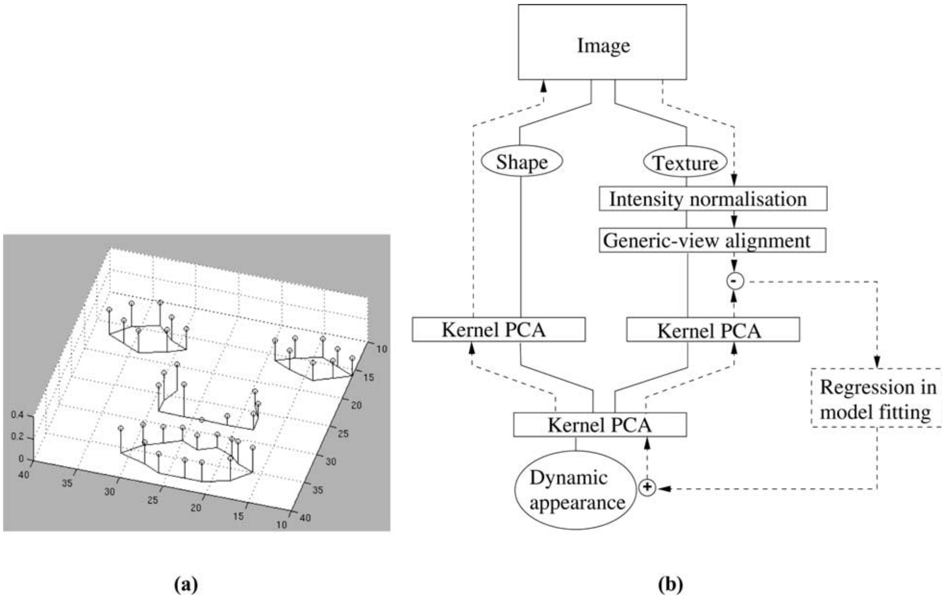

Fig. 6. (a) Variance of the error (confidence measure) from inner landmark alignment across pose. The *z*-axis of this 3D graph is proportional to the distance covered by an inner landmark on the aligned shape as the pose varies from profile to profile. Ideally this distance for all landmarks should be null, as it is for the nose-tip. (b) Computations for constructing a dynamic appearance model across views. The projection to and back-projection from the model are outlined in plain line. The model fitting process to novel images is outlined in dashed line.

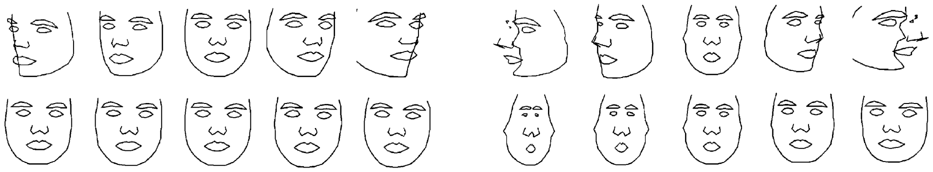

Fig. 7. The first (top) and second (bottom) modes of shape variation for a linear PDM covering 50° views (left) and across 180° views (right). The valid shape range (VSR) for training the ±20° PDM was set to $\pm 3\sqrt{\lambda_i}$ and for across ±90° views, it was limited to $\pm 0.2\sqrt{\lambda_i}$.

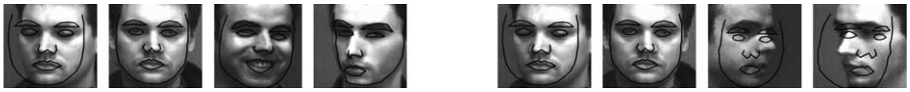

Fig. 8. Fitting shapes to images using linear ASMs trained across ±20° (left) and ±90° (right).

6. Experiments

To illustrate our approach for corresponding dynamic appearances across views, we use a face database composed of images of six individuals taken at pose angles ranging from -90 to $+90°$ at 10° increments. The pose of a face is labelled by means of an electro-magnetic sensor attached to the subject's head and a camera calibrated relative to the transmitter [19,32]. The electro-magnetic sensor has an average angular error of 4–6° in both tilt and yaw. The landmarks on the training faces were manually located for training.

On linear ASMs coping with pose change. A linear PDM trained to capture face shape variation between narrow views (±20°) was compared to a PDM trained for a full range of poses between ±90°. Fig. 7 shows the two main modes of variation for each of these linear PDMs.

The two PDMs in Fig. 7 and their corresponding LGL models were used to fit ASMs to face images, as shown in Fig. 8. Using the model trained for the 50° pose range, an ASM was able to fit shapes to face images quite well (left). However, when the PDM across the view-sphere was used, an ASM was only able to fit shape satisfactorily near the frontal view. At most of the other poses, the ASM was unable to recover the shape within the VSR.

On view-context based non-linear ASM. In comparison, KPCA was used to train a non-linear PDM and capture face shape variation across views (±90°). Fig. 9 shows the three

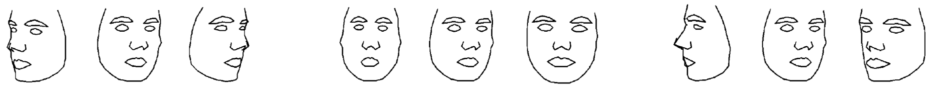

Fig. 9. First three modes of shape variation for a KPCA based non-linear PDM. The VSR was set to $-1.5\sqrt{\lambda_i} \le b_i \le 1.5\sqrt{\lambda_i}$.

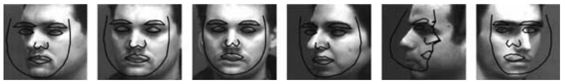

Fig. 10. Fitting shapes to images at different views using a non-linear ASM.

main modes of variation and that the non-linear PDM succeeds in extending shape VSR where the linear PDMs had failed, as shown in Fig. 7.

The non-linear PDM and its corresponding LGL models were used to fit a non-linear ASM on face images, as shown in Fig. 10. The non-linear ASM converges and recovers shapes within the VSR but not to the right shape. This is because sometimes the background grey-levels are very similar to the grey-levels around certain landmarks at specific poses. In such cases, using *local* grey-levels alone will fail to find correspondence between views. To better discriminate object foreground and background, we introduce explicit pose index as view-context based constraint.

A view-context based non-linear PDM and its corresponding LGL models were used to fit novel face images, as shown in Fig. 11. The ASM converges to the right shape and is able to recover the pose. We used the frontal view shape to start fitting. For the first iteration, the landmarks were allowed to move along the normals to the shape contour for up to a distance of 12 pixels on each side.

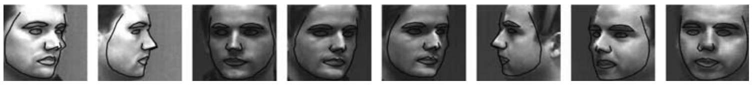

Fig. 11. Fitting shapes to images at different views using a view-context based non-linear ASM.

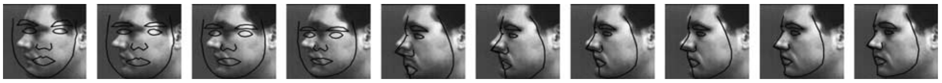

Fig. 12. An example of fitting a shape to a face image and recovering its pose at −80°. The estimated shapes overlaid upon the images are shown after iterations 0, 1, 4, 6, 12, 13, 15, 16, 20 and 25.

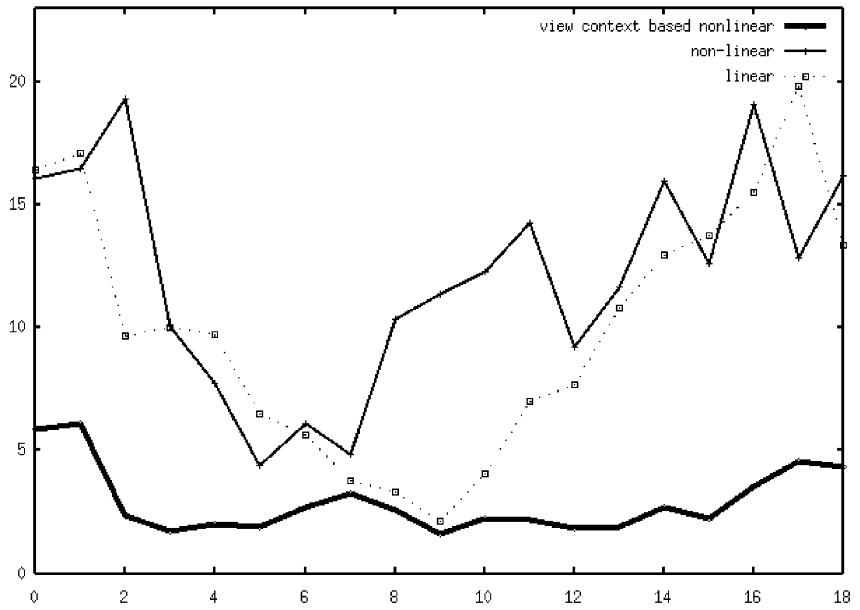

Fig. 13. Comparing shape fitting errors across views. Typical fitting errors (vertical axis) of different ASMs in pixels are drawn against pose in yaw at 20° interval (horizontal axis). Whilst the dashed-line represents a linear ASM, the plain-line is for a non-linear ASM and the bold-line for a view-context based non-linear ASM.

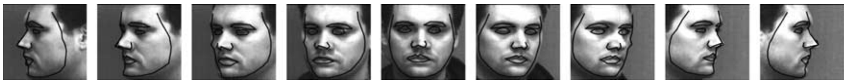

Fig. 14. Examples of fitting shapes to images at novel views using a view-context based non-linear ASM.

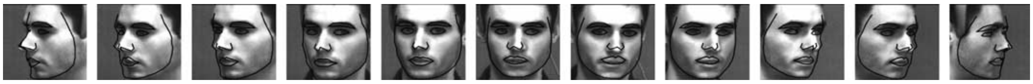

Fig. 15. An example of fitting shapes to images of an unknown face across views using a view-context based non-linear ASM.

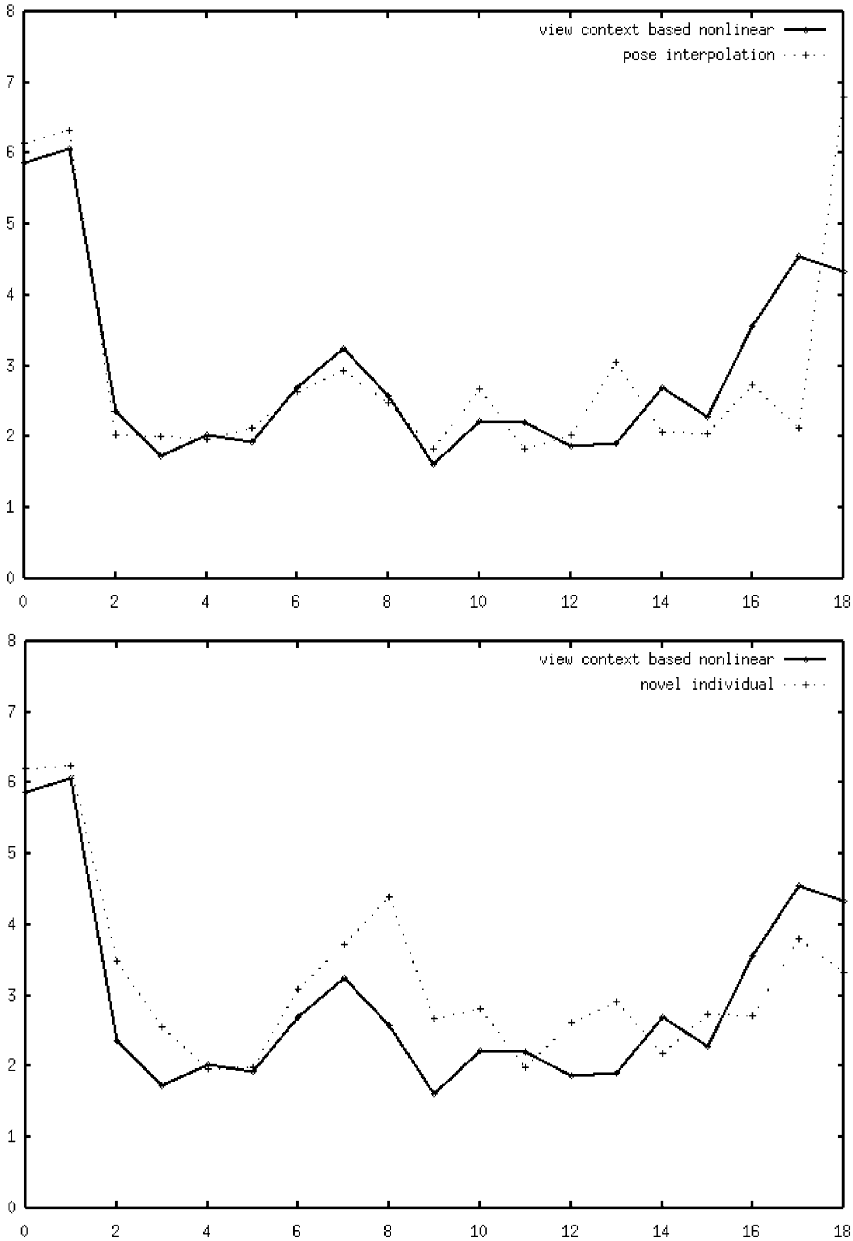

Fig. 16. Top: A view-context based non-linear ASM trained on all poses in yaw (plain-line) gave similar fitting errors compared to a model trained only on half of the poses in yaw (dashed-line). Bottom: Similar fitting errors also exist for a model trained on all faces (plain-line) and a model trained only on some of the faces and tested on a novel face (dashed-line). The horizontal axis shows the yaw variation at 20° interval and the vertical axis shows fitting errors in pixels.

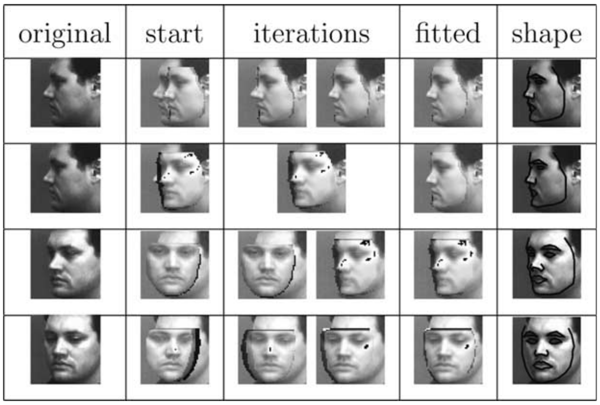

Fig. 17. Each row is an example of both texture and shape fitting. The left image of each row is the original image, the following are the images obtained at successive iterations. The penultimate image shows the converged fitting of both the shape and texture and the last image overlaps the recovered shape on the original image. The first example had pose offset of 20° and translational offset of five pixels in *x*-direction. The second, third and fourth had pose offsets of 40, 50 and 40°, respectively. The width of a face is on average 30 pixels.

This was then adjusted proportionally to the fitting error after each iteration. A LGL model was built using three pixels on both sides of a landmark along the normal to the shape. Both the PDM and the LGLs were restrained to ten-dimensional eigenspaces. Fig. 12 shows an example of fitting a shape to a face image. From left to right, the images depict the shape transformation in the process. Fig. 13 compares fitting errors from different ASMs. A linear ASM performs better at the mean pose than at extreme poses. A non-linear ASM exhibits similar results except at the mean pose. For all poses, a view-context based non-linear ASM performs significantly better.

Generalisation to novel views and novel faces. Two more experiments were conducted to evaluate the capability of the view-context based non-linear ASM for interpolating shape of novel faces not in the training set and recovering poses at novel views. A view-context based non-linear ASM was first trained at 20° pose intervals between ±90°. The model was then used to recover both the shape and pose of faces at novel views. Here the number of eigenvectors was increased to 20 and the VSR was extended to 10 times the standard deviation. Examples of shape fitting at novel views between known poses are shown in Fig. 14.

A view-context based non-linear model was also trained to recover both the shape and pose of novel faces not in the training set. A model was trained on all but one of the faces in a database and was then tested on all poses of an unknown face. The experiment was performed for a number of unknown faces and an example is shown in Fig. 15. A comparison of fitting errors from model generalisation is shown in Fig. 16.

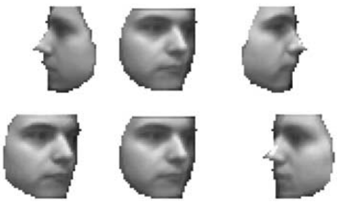

Fig. 18. First two modes of variations of the combined pose-invariant shape and texture appearance model (±3 std).

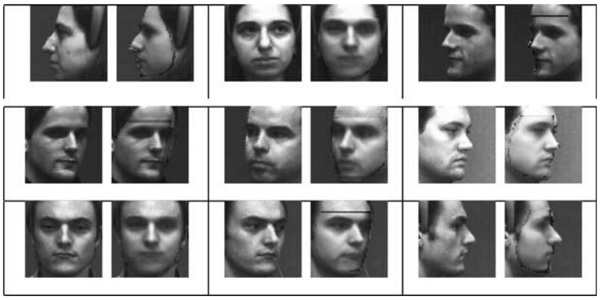

Fig. 19. Each pair of images is an example of texture fitting. The first image is the original image and the second is the texture fitted by the model with known shapes.

On dynamic face appearance models across views. A dynamic face appearance model was first trained using face images of one individual at 19 poses (from −90° to +90° at 10° increments, one image per pose). Four, ten and six eigenvectors were retained to model the shape, the texture and the combined appearance, respectively. Examples of fitting are shown in Fig. 17.

A more generic dynamic face appearance model was then trained using images of faces of five individuals at 19 poses. Ten, 40 and 20 eigenvectors were retained to model the shape, the texture and the combined appearance, respectively. The first two modes of variation are shown in Fig. 18. Fig. 19 shows examples of fitting texture with known

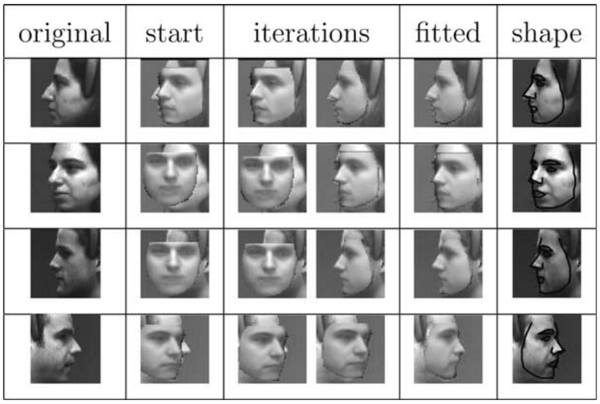

Fig. 20. Each row is an example of both shape and texture fitting. The first image is the original image, the following are the images obtained at successive iterations. The penultimate image shows the converged fitting of both the shape and texture. The last image overlaps the recovered shape onto the original image. The first example started with pose offset of 40°. The second example started with pose offset of 50° and translational offset of −4 pixel in *x*-direction and −3 pixels in *y*-direction. The third and fourth examples started with pose offset of 90 and 40°, respectively, and they both had a translational offset of −6 pixels in *x*-direction.

shapes at different views of large pose variation. Examples in Fig. 20 show model fitting and reconstructions when both shape and texture are unknown. While the shape (both the pose and the feature points) can be recovered adequately, this is no longer the case for texture. Whilst the pose of the texture can be recovered correctly, the intensity information of *all the pixels* are not always recovered accurately. The reason for such effect is that the alignment of all pixel is only approximated and the variation in the aligned texture due to the pose change overwhelms the variation due to identity difference. It is worth pointing out though that for recognition purposes, accurate texture reconstruction at every pixel may not be required [18,39].

7. Conclusions

In this work, we have focused on the problem of modelling the 2D dynamic appearance of faces across significantly different views. Appearance of a face varies considerably across views, more so than those of different faces at the same pose. Such variations are intrinsically non-linear. This non-linearity makes establishing accurate correspondence difficult. A central computational issue of concern is how knowledge of both shape and texture of faces at a single or a set of familiar view(s) can be generalised to novel views. In particular, we considered how structural knowledge about shape could be learned and used to provide the necessary correspondence for obtaining approximately shape-free texture across views.

We addressed the problem by learning non-linear shape transformation across views using Kernel PCA based on SVM. We also augmented the non-linear 2D active shape model with pose constraint. We further presented a method for constructing a *dynamic face appearance model* able to capture both the shape and the texture of faces from profile to profile views. The non-linearities of such variations were again learned using Kernel PCA. In order to bootstrap the alignment and warp texture, a generic-view 2D shape template was introduced.

So far, different views of faces have been treated as a collection without any order. The problem of learning transformation functions of face appearances between views may be made unnecessarily hard as a result. Realistically, however, faces are observed continuously in space and *over time*. Consequently, dynamic 2D facial appearances across views ought to be spatio-temporally continuous and progressive. Our current and future work focuses on addressing the following question: to what extent the temporal continuity and the *ordering* of different views can be exploited in extracting knowledge about facial appearances over time.

References

[1] D. Beymer, Feature correspondence by interleaving shape and texture computations, IEEE CVPR, 1996, pp. 921–928.

[2] D. Beymer, T. Poggio, Image representations for visual learning, Science 272 (1996) 1905–1909.

[3] M.J. Black, A.D. Jepson, Eigentracking: robust matching and tracking of articulated objects using a view-based representation, ECCV, vol. 1, Cambridge, UK, 1996, pp. 329–342.

[4] R. Bowden, T.A. Mitchell, M. Sahardi, Cluster based non-linear principal component analysis, IEE Electronics Letters 33 (22) (1997) 1858–1859.

[5] R. Bowden, T.A. Mitchell, M. Sarhadi, Reconstructing 3d pose and motion from a single camera view, BMVC, Southampton, 1998, pp. 904–913.

[6] T. Cootes, G. Edwards, C. Taylor, Active appearance models, ECCV, Freiburg, 1998, pp. 484–498.

[7] T. Cootes, A. Hill, C. Taylor, J. Haslam, The use of active shape models for locating structures in medical images, Image and Vision Computing 12 (1944) 355–366.

[8] T. Cootes, C. Taylor, A mixture model for representing shape variation, Image and Vision Computing 17 (1999) 567–573.

[9] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active shape models—their training and application, Computer Vision and Image Understanding 61 (1) (1995) 38–59.

[10] N. Costen, I. Craw, G. Robertson, S. Akamatsu, Automatic face recognition: what representation?, ECCV, Cambridge, UK, 1996, pp. 504–513.

[11] I. Craw, Machine coding of human faces, Technical report, Department of Mathematical Sciences, University of Aberdeen, 1996.

[12] I. Craw, P. Cameron, Face recognition by computer, BMVC, Leeds, UK, 1993, pp. 489–507.

[13] F. de la Torre, S. Gong, S. McKenna, View-based adaptive affine tracking, ECCV, vol. 1, Freiburg, Germany, 1998, pp. 828–842.

[14] G. Edwards, A. Lanitis, C. Taylor, T. Cootes, Modelling the variability in face images, IEEE Face and Gesture Recognition, Killington, Vermont, 1996, pp. 328–333.

[15] A. Gee, R. Cipolla, Determining the gaze of faces in images, Image and Vision Computing 12 (10) (1994) 639–647.

[16] A. Gee, R. Cipolla, Fast visual tracking by temporal consensus, Image and Vision Computing 14 (2) (1996) 105–114.

[17] S. Gong, J.M. Brady, Parallel computation of optic flow, ECCV, Antibes, France, April 1990, pp. 124–134.

[18] S. Gong, S. McKenna, J.J. Collins, An investigation into face pose distributions, IEEE Face and Gesture Recognition, Killington, Vermont, 1996, pp. 265–270.

[19] S. Gong, E.-J. Ong, S. McKenna, Learning to associate faces across views in vector space of similarities to prototypes, BMVC, Southampton, 1998, pp. 54–63.

[20] T. Heap, D. Hogg, Improving specificity in pdms using a hierarchical approach, BMVC, Colchester, 1997, pp. 80–89.

[21] N. Kruger, M. Potzsch, T. Maurer, M. Rinne, Estimation of face position and pose with labeled graphs, BMVC, Edinburgh, 1996.

[22] A. Lanitis, C. Taylor, T. Cootes, Automatic interpretation and coding of face images using flexible models, IEEE PAMI 19 (7) (1997) 743–756.

[23] A. Lanitis, C. Taylor, T. Cootes, T. Ahmed, Automatic interpretation of human faces and hand gestures using flexible models, IEEE Face and Gesture Recognition, Zurich, 1995, pp. 98–103.

[24] S. McKenna, S. Gong, Real-time face pose estimation, Journal of Real-Time Imaging, Special Issue on Real-Time Visual Monitoring and Inspection 4 (1998) 333–347.

[25] S. McKenna, S. Gong, R.P. Würtz, J. Tanner, D. Banin, Tracking facial feature points with gabor wavelets and shape models, IAPR International Conference on Audio-Video Based Biometric Person Authentication, Crans-Montana, Switzerland, 1997, pp. 35–43.

[26] S. Mika, B. Schölkopf, A. Smola, G. Ratsch, K. Muller, M. Scholz, G. Ratsch, Kernel pca and de-noising in feature spaces, NIPSS, 1998.

[27] J. Ng, S. Gong, Learning support vector machines for a multi-view face model, BMVC, vol. 2, Nottingham, UK, September 1999, pp. 503–512.

[28] E.-J. Ong, S. Gong, A dynamic human model using hybrid 2d–3d representations in hierarchical pca space, BMVC, Nottingham, 1999.

[29] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, IEEE CVPR, Seattle, 1994, pp. 84–91.

[30] T. Poggio, S. Edelman, A network that learns to recognise three-dimensional objects, Nature 343 (1990) 263–266.

[31] B. Schölkopf, A. Smola, K. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.

[32] J. Sherrah, S. Gong, E.-J. Ong, Understanding pose discrimination in similarity space, BMVC, vol. 2, Nottingham, UK, 1999, pp. 523–532.

[33] S. Ullman, High-Level Vision: Object Recognition and Visual Cognition, MIT Press, Cambridge, MA, 1996.

[34] S. Ullman, R. Basri, Recognition by linear combinations of models, IEEE PAMI 13 (10) (1991) 992–1006.

[35] V. Vapnik, The nature of statistical learning theory, Springer, Berlin, 1995.

[36] T. Vetter, T. Poggio, Linear object classes and image synthesis from a single example image, IEEE PAMI 19 (7) (1997) 733–742.

[37] P.A. Viola, W.M. Wells, Alignment by maximization of mutual information, IEEE ICCV, MIT, MA, 1995, pp. 16–23.

[38] L. Wiskott, Labeled graphs and dynamic link matching for face recognition and scene analysis, PhD thesis, Ruhr-Universität Bochum, Germany, 1995.

[39] R.P. Würtz, Multilayer Dynamic Link Networks for Establishing Image Point Correspondences and Visual Object Recognition, Vol. 41 of Reihe Physik, Verlag Harri Deutsch, Thun, Frankfurt am Main, 1995.